

## **REMARKS/ARGUMENTS**

### **Status of Claims**

Prior to entry of this amendment, the application included claims 1 – 6 and 8 – 25. Claims 1, 13, and 25 are amended; no claims are cancelled or added. Therefore, claims 1 – 6 and 8 – 25 remain present for examination, and claims 1, 13, and 25 are the independent claims. Claims 1 – 6 and 8 – 25 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. Claims 1 – 6 and 8 – 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the cited portions of U.S. Patent Publication No. 2002/0129129 to Bloch et al. (“Bloch”), and further in view of the cited portions of U.S. Patent No. 6,874,143 to Murray et al. (“Murray”). Applicant traverses at least for the reasons below.

### **Rejections Under 35 U.S.C. § 112**

Claims 1 – 6 and 8 – 25 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. Examiner seems to contend that the program resident on the client computer would be used to receive the plurality of text files defining the component of the application, and must, therefore, be executed prior to the receipt of those files. Applicant respectfully traverses this rejection, at least because (1) another program could receive the text files prior to the execution of “the program” recited in the claims; or (2) “the program” could be used to receive the text files at some point and then, subsequently, executed for the functionality recited in the claims.

However, for the sake of furthering prosecution, independent claims 1, 13, and 25 are amended in this Amendment at least to address Examiner’s concerns with regards to enablement. In light of these amendments, Applicant believes the § 112 rejections are moot, and Applicant respectfully requests their withdrawal.

### **Rejections Under 35 U.S.C. § 103(a)**

Independent claims 1, 13, and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Bloch, and further in view of Murray. To establish a *prima facie* case of obviousness, the cited references, combined with the knowledge of those of ordinary skill in the

art, must teach or suggest all the claim limitations. Applicant respectfully traverses this rejection at least because the combination of the cited references with ordinary knowledge in the art fails to teach or suggest all the recitations of claims 1, 13, and 25.

The Office Action's responses to Applicant's previous arguments all seem based on a broad reading of the recitation "executing said application on said runtime environment independent from said program and independent from the program interacting with the server" found in each of the independent claims. Apparently, the Office Action has broadly construed the term "independently" to read the claim recitations as functioning "independently in varying degrees. However, Applicant believes that none of these constructions is consistent with what was intended. As such, claims 1, 13, and 25 are amended in this Amendment to further recite that the application is executed "independently..., such that said application remains executable on said runtime environment upon removal of said program from said client computer system."

Therefore, Applicant maintains his previously-submitted responses to the § 103 rejections, and respectfully requests that they be reconsidered in light of the claim amendments submitted herein.

Specifically, Bloch and Murray do not teach or suggest "executing said application on said runtime environment independent from said program and independent from the program interacting with the server, such that said application remains executable on said runtime environment upon removal of said program from said client computer system," as recited in the independent claims. First, the AVM in Bloch apparently constructs virtual applications "on the fly," or "dynamically," by maintaining contact with web servers and databases (see, e.g., paragraphs [0032], [0034], [0037], [0038], [0047], [0058], [0060], etc.). As such, the application is executed dependently, not independently, on network interactions between the AVM and the server.

Further, Bloch fails to teach a program that builds an application that is resident and executable on the client computer, such that said application remains executable on said runtime environment upon removal of said program from said client computer system, as recited in the claims as amended. Bloch apparently discloses a program (i.e., an Application Virtual Machine, or AVM) that "downloads one or more text files from the server, retrieves program

logic from each of the downloaded files, and assembles the retrieved program logic into a functioning, graphical application in temporary memory" (see Bloch, Abstract (emphasis added)). Thus, essentially, Bloch seems to teach a program that only temporarily invokes an application, where the application is executable only within the program and runtime (the AVM) that invoked it (i.e., the application is spawned in RAM as a part of the AVM).

For example, according to Bloch, a user may run an application by accessing the AVM, waiting for the AVM to assemble the application in temporary memory from server-level application and data files, and interacting with the application through the AVM's runtime environment. The application apparently becomes merely a temporary function of the AVM, which may not be executed outside the AVM. Further, if the AVM in Bloch is deleted, the spawned "application" could no longer be accessed (i.e., the application could not be re-assembled for execution without reloading the AVM). This is different from using an agent program to compile an application so the application may be executable independent from the agent program in a runtime environment independent from the agent program. Unlike in Bloch, if the agent program is removed, the independent application may still be retained and executed in a wholly independent runtime environment.

Many portions of Bloch seem to support this understanding of Bloch as generally teaching that the application execution is performed by, and completely dependent on, the AVM. For instance, paragraph [0028] of Bloch describes the AVM as providing "a method for deploying and executing Extensible Markup Language applications . . ." Paragraphs [0030] – [0034] and [0044] include similar statements. Most striking, Paragraph [0047] states that "the AVM is used to execute 'Virtual Applications' that are assembled from files and programs residing on one or more computers on a network. Only AVM 221 is installed on a Client Device 10. Other files residing on Network Server(s) 131 are downloaded over a Network 121 to assemble the complete Virtual Application 20 that is then executed by the AVM 221 on the Client Device 10." Paragraph [0047] goes on to note that "[t]he XML Files 144 and Image Files 145 specify the appearance and behavior of the assembled Virtual Application 20 during execution by the AVM 221 and in response to user interactions during such execution."

The portions of Bloch cited by the Office Action even further seem to support Applicant's reading of Bloch. For example, the Office Action points to paragraphs [0086] and [0087] as teaching an application that is executable independent from the program that created the application. However, Paragraph [0086] describes tasklists that include client tasks, conditional tasks and host tasks, and Paragraph [0087] states that these "[t]asklists are executed by the AVM when an event occurs to which they are linked." A tasklist cannot be both executed by an AVM and executable independent from the AVM.

The Office Action also points to paragraph [0100]. This paragraph describes the interaction of the application and the AVM. For example, paragraph [0100] describes "[a]s the user interacts with components, the AVM 221 detects, in a platform dependent way, changes to the visual appearance of the component that reflect the entry of data, the selection of a choice, putting focus on a component, and the like (step 92)." The paragraph also describes how an AVM session is ended after function is executed by the system handler. If the function (or application) is truly independent of the AVM, why then is the AVM terminated at the end of the function? Thus, the application and AVM are dependent applications rather than independent applications.

The Office Action also points to paragraph [0102] as describing message boxes that are unique to the operating system. These message boxes occur in response to some action by the AVM. Thus, these message box tasks are dependent on the AVM and not an application created by the AVM. Moreover, these message boxes are, as noted by the Office Action, operating system procedures that are not an application created by the AVM from downloaded text files. The claims require that the same application that is created by the AVM execute independent from the AVM. Operating system procedures or calls have not been downloaded by the AVM.

The Office Action also points to paragraph [0103] as describing remote procedures. Remote procedures are procedures that occur remote from the client computer system and not at the client computer system. The claims require, in part, that a program (i.e., the AVM) create an application from text files on the client computer system and that the application is executable independent from the program. Simply put, remote calls are not

executed on the client computer system. Moreover, these remote calls are not part of the application downloaded and created by the AVM.

The Office Action also points to paragraph [0105] of Bloch as referring to a database handler. This paragraph does not teach or suggest that the database handler is executable independent of the AVM. There is also nothing in Bloch to suggest that the database handler is created by the AVM from text files.

The Office Action also points to paragraph [0109] of Bloch as referring to “user interface . . . accessing any database . . . to test new software.” Paragraph [0109] describes how the AVM is used to test software, assemble user interfaces, access any database, etc. Therefore, the AVM as described is not independent from anything. It is completely dependent with software testing, user interface assembling, and database accessing.

From these and other exemplary portions of Bloch, it is clear that the application is not executable independent from the AVM or independent of server interactions. The Office Action states that: “The paragraphs proffered by Applicants are exactly not utilized of the current Office Action addressing the ‘independent from’ limitation. The language regarding ‘independent from said program’ has been construed using broad reasonable interpretation....” Office Action, pp. 12 – 13. While the Office Action may not specifically cite these paragraphs as teaching this specific recitation of the claims, they are instructive for at least two reasons. First, in light of the amendments to the claims, Applicant contends that the “broad reasonable interpretation” of the Office Action is no longer reasonable and should be withdrawn. Second, the paragraphs of Bloch illustrate that throughout its disclosure, Bloch appears to teach away from the type of client-server and program-application interactions recited in the independent claims.

For at least these reasons, Bloch fails to teach or suggest all the recitations of independent claims 1, 13, and 25. Further, the Office Action provides no suggestion that Murray or ordinary knowledge in the art includes any teachings to remedy these deficiencies of Bloch. As such, the combined teachings of the art, as cited by the Office Action, fail to establish a *prima facie* case of obviousness as to independent claims 1, 13, and 25. Moreover, claims 2 – 6, 8 – 12, and 14 – 24 each depend from these independent claims, and are believed allowable for at

least the same reasons as given above. Applicant, therefore, respectfully requests that the §103(a) rejections to these claims be withdrawn.


**CONCLUSION**

In view of the foregoing, Applicant believes all claims now pending in this application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 303-571-4000.

Respectfully submitted,

Date: February 20, 2009

  
Daniel J. Sherwinter  
Reg. No. 61,751

TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, Eighth Floor  
San Francisco, CA 94111-3834  
Tel: 303-571-4000  
Fax: 415-576-0300  
DJS/sk  
61780017 v1